

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-175353
 (43)Date of publication of application : 02.07.1999

(51)Int.Cl. G06F 9/46
 G06F 1/14

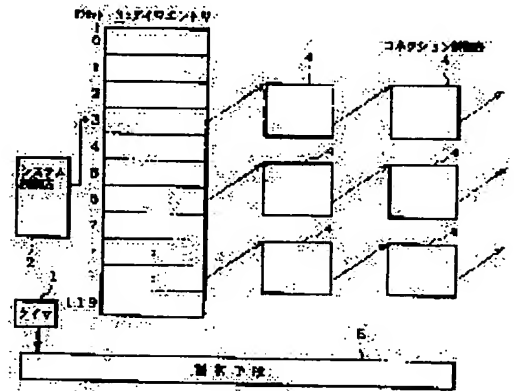
(21)Application number : 09-338799 (71)Applicant : FUJITSU LTD
 (22)Date of filing : 09.12.1997 (72)Inventor : YAMAZAKI TAKESHI

(54) TIMER CONTROLLER AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To lighten the load and to reduce the memory consumption by eliminating a count-down process at the time of the timer process of many connections by enqueueing a control table in a timer entry which enters a time-out state while made to correspond to one timer and a timer interval and performing a time-out process for connections in the control table together.

SOLUTION: The connection control table 4 is enqueueed in the timer entry 3 corresponding to the time when a time-out report is requested. The timer entry 3 circulates and moves in order according to the system control table 2 by reports from the timer 1 which show a specific time has been elapsed and when the connection control table 4 is enqueueed in the timer entry 3, a time-out report is sent to the corresponding connections (timer request source) according to the connection control table 4. Consequently, automatic information becomes possible only by enqueueing the connection control table 4 in the timer entry 3 of desired time.



LEGAL STATUS

[Date of request for examination]
 [Date of sending the examiner's decision of rejection]
 [Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
 [Date of final disposal for application]
 [Patent number]
 [Date of registration]
 [Number of appeal against examiner's decision of rejection]
 [Date of requesting appeal against examiner's decision of rejection]
 [Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The timer control unit which is characterized by providing the following and which controls much timer demands. One timer. The timer entry of the circulated predetermined number which was matched and was prepared in the time interval of the one above-mentioned timer. The control table which carries out a queuing to the timer entry corresponding to time with the timer demand of the timer entries of the above-mentioned predetermined number which carried out circulation. A means to notify a time-out to the timer demand origin of the control table concerned when the ENQ of the above-mentioned control table is carried out to the timer entry which circulates through for every notice from the one above-mentioned timer and which carries out the point one by one.

[Claim 2] The timer control unit according to claim 1 characterized by carrying out two or more sequential queuing of the above-mentioned control table to the above-mentioned timer entry.

[Claim 3] The claim 1 characterized by setting two or more timer demands as the above-mentioned control table, carrying out the queuing of the control table concerned to the above-mentioned timer entry corresponding to the shortest time-out time, and carrying out the re-queuing of the control table concerned to the above-mentioned timer entry corresponding to the shortest next time-out time when a time-out occurs, or a timer control unit according to claim 2.

[Claim 4] The claim 1 characterized by updating until it sets up the residual time from the present time, it carries out a queuing to a timer entry and residual time becomes a timer demand beyond the time expressed by the timer entry of the above-mentioned predetermined number which carried out circulation below at 1 period time part at the time of processing of the timer entry concerned, or one of timer control units according to claim 3.

[Claim 5] A means which was matched and was prepared in the time interval of one timer to access the timer entry of the circulated predetermined number, The means which carries out the queuing of the control table to the timer entry corresponding to time with the timer demand of the timer entries of the circulated predetermined number, The record medium which recorded the program operated as a means to notify a time-out to the timer demand origin of the control table concerned when the ENQ of the above-mentioned control table is carried out to the timer entry which circulates through for every notice from the one above-mentioned timer, and which carries out the point one by one and in which computer reading is possible.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] this invention relates to the timer control unit and record medium which control much timer demands.

[0002]

[Description of the Prior Art] When processing a lot of connections' timer conventionally, there is the technique of processing all timer demands with one timer.

[0003] Moreover, it has a timer for every connection and there is the technique of processing a timer.

[0004]

[Problem(s) to be Solved by the Invention] When it was going to continue at once and was going to deal with count-down processing of the timer which will be started through each connection if the one former timer mentioned above performs timer processing from all connections, and time-out processing, the continuation occupancy time of CPU became long and there was a problem that it will be kept waiting even if there is interruption between them.

[0005] Moreover, even when it had a timer for every connection of the latter mentioned above, the processing which repeats a count-down for every predetermined time was needed, respectively, and there was a problem that the rate kept waiting even if there is other interruption too, when the number of connections becomes abundant became high.

[0006] In order that this invention may solve these problems, it prepares one timer, matches it with a timer interval, it carries out the queuing of the control table to the timer entry which prepares and carries out the time-out of two or more timer entries, summarizes time-out processing on the connection of the control table concerned, gives it to him, loses each count-down processing at the time of timer processing of a lot of connections, and it aims at cutting down memory consumption while it mitigates a load sharply.

[0007]

[Means for Solving the Problem] With reference to drawing 1, a The means for solving a technical problem is explained. In drawing 1, a timer 1 is one timer.

[0008] The timer entry 3 is a timer entry of the circulated predetermined number which was matched and was prepared in the time interval of one timer. The queuing of the connection control table 4 is carried out to the timer entry 3 corresponding to time with the timer demand of the timer entries 3 of the circulated predetermined number.

[0009] Control means 5 control. Next, operation is explained. When the ENQ of the connection control table 4 is carried out to the timer entry 3 which circulates through for every notice from one timer 1 and which carries out the ENQ of the connection control table 4 to the timer entry corresponding to time with the timer demand of the timer entries 3 of the predetermined number which control means 5 circulated, and carries out the point one by one, it is made to notify a time-out to the timer demand origin of the connection control table 4 concerned.

[0010] Under the present circumstances, it is made to carry out two or more sequential queuing of the connection control table 4 to the timer entry 3. Moreover, two or more timer demands are set as the connection control table 4, and the queuing of the connection control table 4 concerned is carried out to the timer entry 3 corresponding to the shortest time-out time, and when a time-out occurs, it is made to carry out the re-queuing of the connection control table 4 concerned to the timer entry 3 corresponding to the shortest next time-out time.

[0011] Moreover, it is made to update until it sets the residual time from the present time as the connection control table 4, it carries out a queuing to the timer entry 3 and residual time becomes a timer demand beyond the time expressed by the timer entry 3 of the circulated predetermined number below at 1 period time part at the time of processing of the timer entry 3.

[0012] Therefore, by preparing one timer, matching with a timer interval, carrying out the queuing of the connection control table 4 to the timer entry 3 which prepares and carries out the time-out of two or more timer entries 3, and carrying out to the connection of the connection control table 4 concerned by summarizing time-out processing, each count-down processing at the time of timer processing of a lot of connections is lost, and while mitigating a load sharply, it becomes possible to cut down memory consumption.

[0013]

[Embodiments of the Invention] Next, the gestalt of operation of this invention and operation are explained to a detail one by one using drawing 6 from drawing 1.

[0014] Drawing 1 shows the system configuration view of this invention. In drawing 1, a timer 1 is one timer and notifies a time-out signal to control means 5 for every (every [for example,] second) predetermined-time interval.

[0015] The system control table 2 sets up and manages the information which circulates through and carries out the point of the timer entry 3 corresponding to the present time of a timer entry, as shown in (a) of drawing 2 mentioned later, and (b) (it mentions later using drawing 2).

[0016] The timer entry 3 is matched with the predetermined-time interval (for example, 1 second) of a timer 1, and are prepared one by one. two or more (for example, 120 pieces)] The connection control table 4 sets up and manages the timer demand information from a connection (timer demand origin).

[0017] Control means 5 perform various control (it mentions later using drawing 3 or drawing 6). When the timer entry 3 which carries out the point circulates, it moves one by one and the ENQ of the connection control table 4 is carried out to the timer entry 3 concerned by the system control table 2 for every notice of the purport which constitutes as mentioned above, carries out the ENQ of the connection control table 4 to the timer entry corresponding to time to require the notice of a time-out, and carried out predetermined-time progress from the timer 1, the connection (timer demand origin) who corresponds based on the connection control table 4 concerned is notified of the purport of a time-out. The purport of a time-out can be notified that the time comes automatically only by carrying out the ENQ of the connection control table 4 to

the timer entry 3 of time to wish by this, and in this case, it is one timer and it becomes possible to notify a time-out without the KAUNDO down processing for every timer demand origin moreover.

[0018] Drawing 2 shows 1 example block diagram of this invention. (a) of drawing 2 shows the concrete block diagram of drawing 1. In (a) of drawing 2, the system control table 2 sets up and manages the following information, as shown in (b) of drawing 2 mentioned later.

[0019] - system timer entry region pointer (STP): - present pointer offset (CPO): - number (STO) of - system timer entries: - under - processing -- connection control-table (ECP): -- here, the system timer entry region pointer STP is a pointer which carries out the point of the head of the field of the timer entry 3. The present pointer offset CPO manages the information which carries out the point of the timer entry to the present time, and is carried out +one for every signal of the predetermined-time interval from a timer 1 here. The address of the actual timer entry 3 is [CPO] x4 from a head (since a timer entry is 4 bytes, 4 is doubled four). The number STO of system timer entries is a total (the example of (a) of drawing 2 120) of a timer entry. The connection control table ECP sets up and manages during processing the information (address) which carries out the point of the connection control table 4 under processing.

[0020] The timer entry 3 prepares 120 to 0-119 like illustration here, and each timer entry is 4 bytes. The predetermined-time interval which has a notice from one timer 1 here will have an entry within a time for 120 seconds in 1 second, then the timer entry 3 at intervals of 1 second. An ENQ is carried out to the timer entry 3 which corresponds when 120 seconds equivalent to 1 round are subtracted and it has been 120 or less seconds, whenever it set residual time as the connection control table 4 and carried out it 1 round so that it might mention later when 120 seconds was exceeded, and a time-out is notified to timer demand origin at the time of the timer entry.

[0021] The connection control table 4 sets up and manages the information on the timer demand from a connection (timer demand origin) (it mentions later using (c) of drawing 2). (b) of drawing 2 shows the example of a system control table. The information mentioned above is set to this system control table 2.

[0022] (c) of drawing 2 shows the example of a connection control table. The following information on illustration is set up and managed in this connection control table 4.

- connection [degree] control-table pointer (NEXTP): - timer during-starting offset (TOF) *1: and timer 1 desired value (T1R): - timer 1 residual value (T1C): -- here, the following connection control-table pointer NEXTP is a pointer when linking the following connection control table 4. The timer during-starting offset TOF sets up and manages the offset value when starting a timer. Timer 1 desired-value T1R sets timer desired value as timer during starting (T1R=0 means un-starting). Timer 1 residual value T1C sets up the residual value of the timer ended late, when two or more timers are started (T1C=0 is set up when there is no residual value).

[0023] Next, according to the sequence shown in the flow chart of drawing 3 or drawing 6, operation of the composition of drawing 2 is explained in detail. Drawing 3 shows the explanation flow chart (the 1) of this invention of operation. This is the explanation flow chart of the whole composition of drawing 2 of operation.

[0024] In drawing 3, ST carries out timer interruption generating from a system. In this example, it is assumed that it generates at a 1-second interval (a timer interruption assumes that it generates at control means 5 at a 1-second interval from a timer 1).

[0025] S2 updates the present pointer offset CPO. This carries out the value of the present pointer offset CPO in the system control table 2 of drawing 2 +one, and updates it. S3 asks for the entry which should be processed. This performs STP+CPOx4 ->WK. Namely, - [system timer entry region pointer STP (start address of timer entry 3)] + [present point offset CPO] x4 in the system control table 2 of drawing 2 (4 bytes) ** is also stopped and this is substituted for a work area WK. Here, 1 of the timer entry 3 entry may be 4 bytes.

[0026] S4 distinguishes whether the connection control table 4 is pointed out from the entry which should be processed. This distinguishes whether the point of the connection control table 4 is carried out from the timer entry 3 of the address searched for by S3. In YES, since it became clear to have pointed out the connection control table 4 from the present timer entry 3, processing of S5 or S7 is performed. On the other hand, since the connection control table 4 is not pointed out from the present timer entry 3 in NO and it is not necessary to perform time-out processing, it returns to S1.

[0027] S5 memorizes the connection control table 4 which should be processed (WK->ECP). S6 performs time-out processing of a connection control table (it mentions later using drawing 4).

[0028] S7 asks for the connection control table 4 which should be processed continuously (NEXP->WK of the connection control table 4). And S4 or subsequent ones are repeated. When the present timer entry 3 is updated one by one whenever a timer interruption occurs from a timer 1 (every [for example,] second), and the connection control table 4 is connected with the present timer entry 3 concerned by the above, according to the flow chart of drawing 4 later mentioned based on the connection control table 4 concerned, it enables a notice etc. to make a time-out timer demand origin at a connection. Under the present circumstances, it is not necessary to perform KAUNDO down processing of a timer etc. for every connection of timer demand origin, and it becomes possible about a throughput to cut down mitigation and memory space. It explains to a detail one by one below.

[0029] Drawing 4 shows the explanation flow chart (the 2) of this invention of operation. This is a detailed explanation flow chart when performing time-out processing based on the connection control table 4 of S7 of drawing 3 mentioned already.

[0030] In drawing 4, S11 distinguishes whether there is any timer which this time-out generates. This distinguishes whether there is any timer of T1R!=0 and TnC=0. In addition, it means that T1R!=0 set timer desired value to timer during starting. T1R=0 means un-starting. TnC=0 expresses the purport which is the time of less than 1 round of the timer entry 3 (in the case of TnC!=0, the residual time more than 1 round is set up).

[0031] S12 performs all time-out processings of the timer Tn which this time-out generates (time-out processing of the timer of TnR!=0 and TnC=0 is performed). S13 makes a idle state all the timers that carried out the time-out this time (0 ->TnR).

[0032] It distinguishes whether S14 has another timer during starting (TnR!=0 carries out timer whereabouts distinction). In YES, a timer is processed during starting by S15 or S21. On the other hand, in NO, the dequeue of the connection control table 4 is carried out from the timer entry 3 by S22.

[0033] S15 searches the timer Tx which a time-out next generates (TxC searches the minimum thing). S16 distinguishes whether a time-out generates Tx in 1 round (TnC<120?). In YES, since it proved that Tx carried out time-out generating into 1 round, S17 or S20 is performed. In NO, since it proved that Tx did not carry out time-out generating into 1 round, during all startings, residual time behind 1 round of timers is memorized to a timer (updating) (TmC-120 ->TmC, TxC-120 ->TxC), and it ends by S21.

[0034] Since it proved that S17 carried out time-out generating into 1 round by YES of S16, the residual time at the time of Tx time-out is updated to all the timers Tm during starting of those other than Tx (TnC-TxC->TnC).

[0035] S18 carries out the re-queuing of the connection control table to the timer entry at the time of Tx time-out (if the dequeue of the connection control table is carried out from the present entry, it is -CPO+TxC<120 and it is a queue and CPO+TxC>=120 at the entry of CPO+TxC, a queue is carried out to the entry of CPO+TxC-120).

[0036] S19 updates the residual time of Tx (0 → TxC). S20 updates offset to the present pointer offset during timer starting of a connection control table. And it ends.

[0037] By S11 of a more than, or S22, time-out processing of a connection control table is able to be performed. Drawing 5 and drawing 6 show the explanation flow chart of this invention of operation. These drawing 5 and drawing 6 explain the procedure when starting other timers T1 in the connection control table A of drawing 2 in full detail.

[0038] In drawing 5, S31 is already distinguished in under timer starting (it distinguishes whether either of the T1 R-T4R of (c) of drawing 2 is 0). In YES, since it was already proved that it is under timer starting, S37 of drawing 6 or processing of S46 is performed by **. On the other hand, in NO, since it proved that it was not [timer / be / it] under starting, processing of S32 or S36 is performed.

[0039] S32 takes out the present pointer offset (CPO), and memorizes storage (T1 → T1R) and timer during-starting offset by making a timer value into timer desired value (CPO → TOF).

[0040] S33 is distinguished in a T1 > timer entry total * timer-interruption interval (T1 > it distinguishes in 120 * 1). Since it proved that T1 was larger than a part for 1 round of a timer entry in YES, it progresses to S34. On the other hand, since it proved that T1 was smaller than a part for 1 round of a timer entry in NO, it progresses to S36.

[0041] Since S34 proved that T1 was larger than a part for 1 round of a timer entry by NO of S33, it puts the value of 1 round after into a timer residual value (T1 - 120 * 1 → T1R).

[0042] S35 carries out the queuing of the connection control table to the present pointer offset of a timer entry. And it ends. Since S36 proved that T1 was smaller than a part for 1 round of a timer entry by YES of S33, the queuing of it is carried out to the entry after T1 progress from the present pointer offset CPO of the timer entry of a connection control table (the queuing of the connection control table is carried out to CPO + T1 of a timer entry). And it ends.

[0043] the case where it is not [timer / of the others in the connection control table 4 / be / it] under starting by NO of S31 of more than, S32, or S36 -- a basis / desired value / timer / T1] -- starting -- carrying out the ENQ of the connection control table to an applicable timer entry, or updating the timer residual value in a connection control table **** -- etc. -- it carries out and it becomes possible to carry out timer starting

[0044] In drawing 6, since S37 is YES of S31 of drawing 5 and was already proved that it is under other timer startings, it asks for the timer in which the timer under present starting carries out a time-out first (if there is a thing of TnR != 0 & TnC = 0, the Tn and Tn of the minimum [TnC] when there is nothing will be calculated). Moreover, T1 is memorized to T1R.

[0045] T1 started this time distinguishes S38 in the value of less than 1 round of a timer entry (it distinguishes in T1 < 120). In YES, it progresses S39. In NO, since T1 started this time was proved that it is a value more than 1 round of a timer entry, the T1 residual time at the time of Tn time-out is memorized by S42. Namely, if it is -CPO >= TOF and is T1R - (CPO - TOF) → T1 C-CPO < TOF, either of the T1R - (120 + CPO) (-TOF) → T1C is performed.

[0046] Since T1 proved S39 to be less than 1 round of a timer entry by YES of S38, Tn under present starting is further distinguished in the value of less than 1 round of a ** timer entry (it distinguishes in TnC = 0). In YES, it progresses S40. In NO, processing of S43 or S46 is performed.

[0047] S40 is distinguished in Tn residual time <= T1. Namely, if it is -CPO >= TOF and is CPO - TOF <= T1R or -CPO < TOF, it distinguishes in 120 + CPO - TOF < T1R. In YES, since it proved that Tn residual time was smaller than T1, the T1 residual time at the time of Tn time-out is memorized by S41. Namely, if it is -CPO >= TOF and is T1R + (CPO - TOF) → T1 C-CPO < TOF, either of T1R + (120 - CPO) - TOF → TnC is updated. On the other hand, in NO of S40, processing of S43 or S46 is performed.

[0048] Since it proves that Tn residual time is not smaller than T1 by becoming clear or NO of S40 if Tn under present starting is not the value of less than 1 round in NO of S39, it performs, and S43 is Tn residual time. - T1 is memorized as residual time of Tn. Namely, if it is -CPO >= TOF (CPO - TOF) and is -T1 R → T1 C-CPO < TOF (120 - CPO), one processing of -TOF - T1 R → TnC is performed.

[0049] S44 updates all TmC(s) of Tm, when Tm under other startings exists (TmC + TnC → TmC). S45 carries out the re-queuing of the connection control table to the queue of T1 time-out time. That is, either of the queues is performed at the offset entry of being a queue and CPO + T1R < 120 (CPO + T1R) to the offset entry of being a dequeue and CPO + T1R >= 120 from the present time entry about - connection control table (CPO + T1R - 120).

[0050] S46 updates timer during-starting offset of a connection control table to the present pointer offset (CPO → TOF). the case where it is timer] already under starting with S37 of a more than, or the procedure of S46 -- the renewal of a connection control table -- tying -- changing -- etc. -- it becomes possible to perform processing

[0051]

[Effect of the Invention] Memory consumption is reducible, while losing each count-down processing at the time of timer processing of a lot of connections and mitigating a load sharply, since the composition performed by preparing one timer, matching with a timer interval, carrying out the queuing of the connection control table to the timer entry which prepares and carries out the time-out of two or more timer entries, and summarizing time-out processing based on an applicable connection control table is adopted according to this invention as explained above.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the system configuration view of this invention.

[Drawing 2] It is 1 example block diagram of this invention.

[Drawing 3] It is the explanation flow chart (the 1) of this invention of operation.

[Drawing 4] It is the explanation flow chart (the 2) of this invention of operation.

[Drawing 5] It is the explanation flow chart (the 3) of this invention of operation.

[Drawing 6] It is the explanation flow chart (the 4) of this invention of operation.

[Description of Notations]

- 1: Timer
- 2: System control table
- 3: Timer entry
- 4: Connection control table
- 5: Control means

[Translation done.]

Requested Patent: JP11175353A
Title: TIMER CONTROLLER AND RECORDING MEDIUM ;
Abstracted Patent: JP11175353 ;
Publication Date: 1999-07-02 ;
Inventor(s): YAMAZAKI TAKESHI ;
Applicant(s): FUJITSU LTD ;
Application Number: JP19970338799 19971209 ;
Priority Number(s): ;
IPC Classification: G06F9/46; G06F1/14 ;
Equivalents: ;

ABSTRACT:

PROBLEM TO BE SOLVED: To lighten the load and to reduce the memory consumption by eliminating a count-down process at the time of the timer process of many connections by enqueueing a control table in a timer entry which enters a time-out state while made to correspond to one timer and a timer interval and performing a time-out process for connections in the control table together. **SOLUTION:** The connection control table 4 is enqueueued in the timer entry 3 corresponding to the time when a time-out report is requested. The timer entry 3 circulates and moves in order according to the system control table 2 by reports from the timer 1 which show a specific time has been elapsed and when the connection control table 4 is enqueueued in the timer entry 3, a time-out report is sent to the corresponding connections (timer request source) according to the connection control table 4. Consequently, automatic information becomes possible only by enqueueuing the connection control table 4 in the timer entry 3 of desired time.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-175353

(43) 公開日 平成11年(1999) 7月2日

(51) Int.Cl.⁶
G 0 6 F 9/46
1/14

識別記号
3 1 5

F I
G 0 6 F 9/46
1/04

3 1 5 Z
3 5 2

審査請求 未請求 請求項の数5 OL (全 11 頁)

(21) 出願番号 特願平9-338799

(22) 出願日 平成9年(1997)12月9日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 山崎 毅

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 岡田 守弘

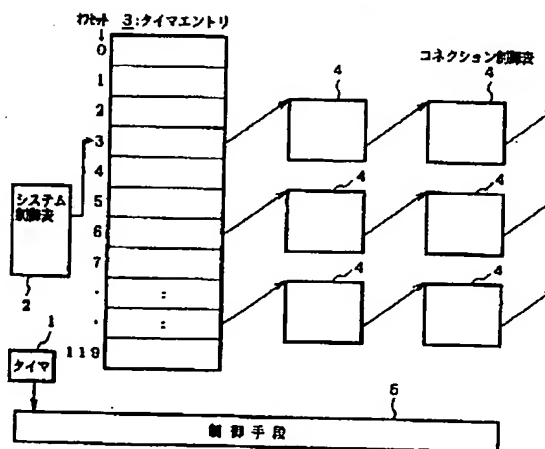
(54) 【発明の名称】 タイマ制御装置および記録媒体

(57) 【要約】

【課題】 本発明は、多数のタイマ要求を制御するタイマ制御装置および記録媒体に関し、1つのタイマを用意し、タイマ間隔に対応づけて複数のタイマエントリを設けてタイムアウトするタイマエントリに制御表をキューイングして当該制御表のコネクションにタイムアウト処理をまとめて行い、多量のコネクションのタイマ処理時のそれぞれのカウントダウン処理を無くし、負荷を大幅に軽減すると共にメモリ消費量を削減することを目的とする。

【解決手段】 1つのタイマと、1つのタイマの時間間隔に対応づけて設けた、循環させた所定個数のタイマエントリと、循環させた所定個数のタイマエントリのうちのタイマ要求のあった時間に対応するタイマエントリにキューイングする制御表と、1つのタイマからの通知毎に循環して順次ポイントするタイマエントリに制御表がエンキューされていたときに制御表のタイマ要求元にタイムアウトを通知する手段とを備えるように構成する。

本発明のシステム構成図



【特許請求の範囲】

【請求項1】多数のタイマ要求を制御するタイマ制御装置において、

1つのタイマと、

上記1つのタイマの時間間隔に対応づけて設けた、循環させた所定個数のタイマエントリと、

上記循環させた所定個数のタイマエントリのうちのタイマ要求のあった時間に対応するタイマエントリにキューイングする制御表と、

上記1つのタイマからの通知毎に循環して順次ポイントするタイマエントリに上記制御表がエンキューされていたときに当該制御表のタイマ要求元にタイムアウトを通知する手段とを備えたことを特徴とするタイマ制御装置。

【請求項2】上記タイマエントリに上記制御表を複数順次キューイングすることを特徴とする請求項1記載のタイマ制御装置。

【請求項3】上記制御表に複数のタイマ要求を設定し、最短のタイムアウト時間に対応する上記タイマエントリに当該制御表をキューイングし、タイムアウトが発生したときに次の最短のタイムアウト時間に対応する上記タイマエントリに当該制御表を再キューイングすることを特徴とする請求項1あるいは請求項2記載のタイマ制御装置。

【請求項4】上記循環させた所定個数のタイマエントリで表現される時間を越えたタイマ要求時に、現在の時間からの残存時間を設定してタイマエントリにキューイングし、当該タイマエントリの処理時に残存時間が1周期時間分以下になるまで更新することを特徴とする請求項1ないし請求項3記載のいずれかのタイマ制御装置。

【請求項5】1つのタイマの時間間隔に対応づけて設けた、循環させた所定個数のタイマエントリにアクセスする手段と、

循環させた所定個数のタイマエントリのうちのタイマ要求のあった時間に対応するタイマエントリに制御表をキューイングする手段と、

上記1つのタイマからの通知毎に循環して順次ポイントするタイマエントリに上記制御表がエンキューされていたときに当該制御表のタイマ要求元にタイムアウトを通知する手段として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】本発明は、多数のタイマ要求を制御するタイマ制御装置および記録媒体に関するものである。

【0002】

【従来の技術】従来、多量の接続のタイマを処理する場合、1つのタイマで全てのタイマ要求を処理する手法がある。

【0003】また、各接続毎にタイマを持ち、タイマを処理する手法がある。

【0004】

【発明が解決しようとする課題】上述した前者の1つのタイマで全ての接続からのタイマ処理を行うと、個々の接続で起動するタイマのカウンタダウン処理、およびタイムアウト処理を、1回に連続して処理しようとする、CPUの連続占有時間が長くなり、その間に割り込みがあっても待たされてしまう問題があった。

【0005】また、上述した後者の接続毎にタイマを持つ場合でも、カウンタダウンを所定時間毎に繰り返す処理がそれぞれ必要となり、接続数が多量になるとやはり他の割り込みがあっても待たされてしまう割合が高くなるという問題があった。

【0006】本発明は、これらの問題を解決するため、1つのタイマを用意し、タイマ間隔に対応づけて複数のタイマエントリを設けてタイムアウトするタイマエントリに制御表をキューイングして当該制御表の接続にタイムアウト処理をまとめて行い、多量の接続のタイマ処理時のそれぞれのカウンタダウン処理を無くし、負荷を大幅に軽減すると共にメモリ消費量を削減することを目的としている。

【0007】

【課題を解決するための手段】図1を参照して課題を解決するための手段を説明する。図1において、タイマ1は、1つのタイマである。

【0008】タイマエントリ3は、1つのタイマの時間間隔に対応づけて設けた、循環させた所定個数のタイマエントリである。接続制御表4は、循環させた所定個数のタイマエントリ3のうちのタイマ要求のあった時間に対応するタイマエントリ3にキューイングするものである。

【0009】制御手段5は、制御を行うものである。次に、動作を説明する。制御手段5が循環させた所定個数のタイマエントリ3のうちのタイマ要求のあった時間に対応するタイマエントリに接続制御表4をエンキューし、1つのタイマ1からの通知毎に循環して順次ポイントするタイマエントリ3に接続制御表4がエンキューされていたときに当該接続制御表4のタイマ要求元にタイムアウトを通知するようにしている。

【0010】この際、タイマエントリ3に接続制御表4を複数順次キューイングするようにしている。また、接続制御表4に複数のタイマ要求を設定し、最短のタイムアウト時間に対応するタイマエントリ3に当該接続制御表4をキューイングし、タイムアウトが発生したときに次の最短のタイムアウト時間に対応するタイマエントリ3に当該接続制御表4を再キューイングするようにしている。

【0011】また、循環させた所定個数のタイマエントリ3で表現される時間を越えたタイマ要求時に、現在の時間からの残存時間をコネクション制御表4に設定してタイマエントリ3にキューイングし、タイマエントリ3の処理時に残存時間が1周期時間分以下になるまで更新するようにしている。

【0012】従って、1つのタイマを用意し、タイマ間隔に対応づけて複数のタイマエントリ3を設けてタイムアウトするタイマエントリ3にコネクション制御表4をキューイングして当該コネクション制御表4のコネクションにタイムアウト処理をまとめて行うことにより、多量のコネクションのタイマ処理時のそれぞれのカウントダウン処理を無くし、負荷を大幅に軽減すると共にメモリ消費量を削減することが可能となる。

【0013】

【発明の実施の形態】次に、図1から図6を用いて本発明の実施の形態および動作を順次詳細に説明する。

【0014】図1は、本発明のシステム構成図を示す。図1において、タイマ1は、1つのタイマであって、所定時間間隔毎（例えば1秒毎）にタイムアウト信号を制御手段5に通知するものである。

【0015】システム制御表2は、後述する図2の(a)、(b)に示すように、タイマエントリの現時刻に対応するタイマエントリ3を循環してポイントする情報などを設定して管理するものである（図2を用いて後述する）。

【0016】タイマエントリ3は、タイマ1の所定時間間隔（例えば1秒）に対応づけて順次複数個（例えば120個）を設けたものである。コネクション制御表4は、コネクション（タイマ要求元）からのタイマ要求情報を設定して管理するものである。

【0017】制御手段5は、各種制御を行うものである（図3ないし図6を用いて後述する）。以上のように構成し、コネクション制御表4をタイムアウト通知を要求する時間に対応するタイマエントリにエンキューしておき、タイマ1からの所定時間経過した旨の通知毎に、システム制御表2によってポイントするタイマエントリ3が循環して順次移動してゆき、当該タイマエントリ3にコネクション制御表4がエンキューされていたときに当該コネクション制御表4をもとに該当するコネクション（タイマ要求元）にタイムアウトの旨を通知する。これにより、希望する時刻のタイマエントリ3にコネクション制御表4をエンキューするのみで、自動的にその時刻になるとタイムアウトの旨の通知することができ、この際に、1つのタイマで、しかもタイマ要求元毎のカウントダウン処理無しにタイムアウトの通知を行うことが可能となる。

【0018】図2は、本発明の1実施例構成図を示す。図2の(a)は、図1の具体的構成図を示す。図2の(a)において、システム制御表2は、後述する図2の

(b)に示すように、下記の情報を設定して管理するものである。

【0019】・システムタイマエントリ域ポインタ(STP)：

・現ポインタオフセット(CPO)：

・システムタイマエントリ数(STO)：

・処理中コネクション制御表(ECP)：

ここで、システムタイマエントリ域ポインタSTPは、タイマエントリ3の領域の先頭をポイントするポインタである。現ポインタオフセットCPOは、現在時刻にタイマエントリをポイントする情報を管理するものであって、ここでは、タイマ1からの所定時間間隔の信号毎に+1するものである。実際のタイマエントリ3のアドレスは、先頭から[CPO]×4である（4はタイマエントリが4バイトであるために4倍したものである）。システムタイマエントリ数STOは、タイマエントリの総数（図2の(a)の例では120）である。処理中コネクション制御表ECPは、処理中のコネクション制御表4をポイントする情報（アドレス）を設定して管理するものである。

【0020】タイマエントリ3は、図示のように、ここでは、0～119までの120個を設けたものであって、各タイマエントリは4バイトである。ここで、1つのタイマ1から通知のある所定時間間隔が1秒とすれば、タイマエントリ3には1秒間隔で120秒分の時間内のエントリを持つこととなる。120秒を越える場合には、後述するように残存時間をコネクション制御表4に設定して1周する毎に1周に相当する120秒を減算して120秒以下になったときに該当するタイマエントリ3にエンキューしてそのタイマエントリのときにタイムアウトをタイマ要求元に通知する。

【0021】コネクション制御表4は、コネクション（タイマ要求元）からのタイマ要求の情報を設定して管理するものである（図2の(c)を用いて後述する）。図2の(b)は、システム制御表2の例を示す。このシステム制御表2には、上述した情報を設定する。

【0022】図2の(c)は、コネクション制御表4の例を示す。このコネクション制御表4には、図示の下記の情報を設定して管理するものである。

・次コネクション制御表ポインタ(NEXTP)：

・タイマ起動時オフセット(TOF)*1：

・タイマ1要求値(T1R)：

・タイマ1残存値(T1C)：

ここで、次コネクション制御表ポインタNEXTPは、次のコネクション制御表4をリンクするときのポインタである。タイマ起動時オフセットTOFは、タイマを起動するときのオフセット値を設定して管理するものである。タイマ1要求値T1Rは、タイマ起動時にタイマ要求値を設定する（T1R=0は、未起動を表す）。タイマ1残存値T1Cは、タイマを複数起動したときに遅く

終了するタイマの残存値を設定するものである（残存値がない場合は $TnC=0$ を設定する）。

【0023】次に、図3ないし図6のフローチャートに示す順序に従い、図2の構成の動作を詳細に説明する。図3は、本発明の動作説明フローチャート（その1）を示す。これは、図2の構成の全体の動作説明フローチャートである。

【0024】図3において、S1は、システムより、タイマ割り込み発生する。本例では、1秒間隔に発生すると仮定する（タイマ1から1秒間隔にタイマ割り込みが制御手段5に発生すると仮定する）。

【0025】S2は、現ポインタオフセットCPOを更新する。これは、図2のシステム制御表2中の現ポインタオフセットCPOの値を+1して更新する。S3は、処理すべきエントリを求める。これは、 $STP+CPO \times 4 \rightarrow WK$

を行う。即ち、図2のシステム制御表2中の

・[システムタイマエントリ域ポインタSTP（タイマエントリ3の先頭アドレス）]+[現ポインタオフセットCPO] $\times 4$ （4バイト）

をもとめ、これをワークエリアWKに代入する。ここで、タイマエントリ3の1エントリは4バイトとする。

【0026】S4は、処理すべきエントリからコネクション制御表4を指しているか判別する。これは、S3で求めたアドレスのタイマエントリ3からコネクション制御表4がポイントされているか判別する。YESの場合には、現タイマエントリ3からコネクション制御表4を指していることが判明したので、S5ないしS7の処理を行う。一方、NOの場合には、現タイマエントリ3からコネクション制御表4を指していない、タイムアウト処理を行う必要がないので、S1に戻る。

【0027】S5は、処理すべきコネクション制御表4を記憶する（ $WK \rightarrow ECP$ ）。S6は、コネクション制御表のタイムアウト処理を行う（図4を用いて後述する）。

【0028】S7は、続けて処理すべきコネクション制御表4を求める（コネクション制御表4の $NEXP \rightarrow WK$ ）。そして、S4以降を繰り返す。以上によって、タイマ1からタイマ割り込みが発生する毎（例えば1秒毎）に、現タイマエントリ3を順次更新して当該現タイマエントリ3にコネクション制御表4がつながれている場合に、当該コネクション制御表4をもとに後述する図4のフローチャートに従い、タイマ要求元にコネクションにタイムアウトを通知などすることが可能となる。この際、タイマ要求元のコネクション毎にタイマのカウントダウン処理などを行う必要がなく、処理量を軽減およびメモリ容量を削減することが可能となる。以下順次詳細に説明する。

【0029】図4は、本発明の動作説明フローチャート（その2）を示す。これは、既述した図3のS7のコネ

クション制御表4をもとにタイムアウト処理を行うときの詳細説明フローチャートである。

【0030】図4において、S11は、今回のタイムアウトの発生するタイマがあるか判別する。これは、 $T1R \neq 0$ 、かつ $TnC=0$ のタイマがあるか判別する。尚、 $T1R \neq 0$ はタイマ起動時にタイマ要求値をセットしたことを表す。 $T1R=0$ は未起動を表す。 $TnC=0$ はタイマエントリ3の1周以内の時間である旨を表す（ $TnC \neq 0$ の場合には1周分以上の残存時間を設定する）。

【0031】S12は、今回のタイムアウトの発生するタイマ Tn の全てのタイムアウト処理を行う（ $TnR \neq 0$ 、かつ $TnC=0$ のタイマのタイムアウト処理を行う）。S13は、今回タイムアウトしたタイマを全て停止状態にする（ $0 \rightarrow TnR$ ）。

【0032】S14は、他に起動中タイマがあるか判別する（ $TnR \neq 0$ のタイマありか判別する）。YESの場合には、S15ないしS21で起動中タイマの処理を行う。一方、NOの場合には、S22でコネクション制御表4をタイマエントリ3からデキューする。

【0033】S15は、次にタイムアウトが発生するタイマ Tx を検索する（ TxC が最小のものを検索する）。S16は、 Tx は1周内にタイムアウトが発生するか判別する（ $TnC < 120?$ ）。YESの場合には、 Tx が1周内にタイムアウト発生すると判明したので、S17ないしS20を実行する。NOの場合には、 Tx が1周内にタイムアウト発生しないと判明したので、S21で全ての起動中タイマにタイマ1周後の残存時間を記憶（更新）し（ $TmC-120 \rightarrow TmC$ 、 $TxC-120 \rightarrow TxC$ ）、終了する。

【0034】S17は、S16のYESで1周内にタイムアウト発生すると判明したので、 Tx 以外の起動中タイマ Tm 全てに対し、 Tx タイムアウト時の残存時間を更新する（ $TnC-TxC \rightarrow TnC$ ）。

【0035】S18は、コネクション制御表を Tx タイムアウト時のタイマエントリに再キューイングする（コネクション制御表を現在のエントリからデキューし、
・ $CPO+TxC < 120$ なら $CPO+TxC$ のエントリにキュー
・ $CPO+TxC \geq 120$ なら $CPO+TxC-120$ のエントリにキューする。

【0036】S19は、 Tx の残存時間を更新する（ $0 \rightarrow TxC$ ）。S20は、コネクション制御表のタイマ起動中オフセットを現ポインタオフセットに更新する。そして、終了する。

【0037】以上のS11ないしS22によって、コネクション制御表のタイムアウト処理が実行できたこととなる。図5および図6は、本発明の動作説明フローチャートを示す。これら図5および図6は、図2のコネクシ

ョン制御表A中の他のタイマT1を起動するときの手順を詳述したものである。

【0038】図5において、S31は、すでにタイマ起動中か判別する（図2の(c)のT1R~T4Rのいずれかが0でないか判別する）。YESの場合には、すでにタイマ起動中と判明したので、①で図6のS37ないしS46の処理を実行する。一方、NOの場合には、タイマ起動中でないと判明したので、S32ないしS36の処理を実行する。

【0039】S32は、現ポイントオフセットを取り出し(CPO)、タイマ値をタイマ要求値として記憶(T1→T1R)、タイマ起動時オフセットを記憶(CPO→TOF)する。

【0040】S33は、 $T1 > \text{タイマエントリ総数} \times \text{タイマ割込間隔}$ か判別する($T1 > 120 \times 1$ か判別する)。YESの場合には、T1がタイマエントリの1周分よりも大きいと判明したので、S34に進む。一方、NOの場合には、T1がタイマエントリの1周分よりも小さいと判明したので、S36に進む。

【0041】S34は、S33のNOでT1がタイマエントリの1周分よりの大きいと判明したので、タイマ残存値に、1周後の値を入れる($T1 - 120 \times 1 \rightarrow T1R$)。

【0042】S35は、コネクション制御表をタイマエントリの現ポイントオフセットにキューイングする。そして、終了する。S36は、S33のYESでT1がタイマエントリの1周分よりの小さいと判明したので、コネクション制御表のタイマエントリの現ポイントオフセットCPOからT1経過後のエントリにキューイングする(コネクション制御表をタイマエントリのCPO+T1にキューイングする)。そして、終了する。

【0043】以上のS31のNO、S32ないしS36によって、コネクション制御表4中の他のタイマ起動中でない場合に、タイマ要求値T1をもとに起動して該当タイマエントリにコネクション制御表をエンキューしたり、コネクション制御表中のタイマ残存値を更新したりなどしてタイマ起動することが可能となる。

【0044】図6において、S37は、図5のS31のYESですでに他のタイマ起動中と判明したので、現在起動中のタイマが最初にタイムアウトするタイマを求める($TnR \neq 0 \& TnC = 0$ のものがあればそのTn、ない場合はTnCが最小のTnを求める)。また、T1をT1Rに記憶する。

【0045】S38は、今回起動するT1がタイマエントリの1周以内の値か判別する($T1 < 120$ か判別する)。YESの場合には、S39に進む。NOの場合には、今回起動するT1がタイマエントリの1周分以上の値と判明したので、S42でTnタイムアウト時のT1残存時間を記憶する。即ち

・ $CPO \geq TOF$ なら $T1R - (CPO - TOF) \rightarrow T1C$

1C

・ $CPO < TOF$ なら $T1R - ((120 + CPO) - TOF) \rightarrow T1C$

のいずれかを実行する。

【0046】S39は、S38のYESでT1がタイマエントリの1周以内と判明したので、更に、現在起動中のTnはがタイマエントリの1周以内の値か判別する($TnC = 0$ か判別する)。YESの場合には、S40に進む。NOの場合には、S43ないしS46の処理を実行する。

【0047】S40は、Tn残存時間 $\leq T1$ か判別する。即ち、

・ $CPO \geq TOF$ なら $CPO - TOF \leq T1R$ か

・ $CPO < TOF$ なら $120 + CPO - TOF < T1R$ か

判別する。YESの場合には、Tn残存時間がT1よりも小さいと判明したので、S41でTnタイムアウト時のT1残存時間を記憶する。即ち、

・ $CPO \geq TOF$ なら $T1R + (CPO - TOF) \rightarrow T1C$

・ $CPO < TOF$ なら $T1R + (120 - CPO) - TOF \rightarrow TnC$

のいずれかの更新を行う。一方、S40のNOの場合には、S43ないしS46の処理を実行する。

【0048】S43は、S39のNOで現在起動中のTnが1周以内の値でないと判明、あるいはS40のNOでTn残存時間がT1よりも小さくないと判明したので、実行するものであって、Tn残存時間-T1をTnの残存時間として記憶する。即ち

・ $CPO \geq TOF$ なら $(CPO - TOF) - T1R \rightarrow T1C$

・ $CPO < TOF$ なら $(120 - CPO) - TOF - T1R \rightarrow TnC$

のいずれかの処理を実行する。

【0049】S44は、その他の起動中のTmが存在する場合、すべてのTmのTmCを更新する($TmC + TnC \rightarrow TmC$)。S45は、コネクション制御表をT1タイムアウト時刻のキューに再キューイングする。即ち、

・コネクション制御表を現在のタイムエントリからデキュー

・ $CPO + T1R \geq 120$ なら $(CPO + T1R - 120)$ のオフセットエントリにキュー

・ $CPO + T1R < 120$ なら $(CPO + T1R)$ のオフセットエントリにキュー

のいずれかを実行する。

【0050】S46は、コネクション制御表のタイマ起動時オフセットを現ポイントオフセットに更新する($CPO \rightarrow TOF$)。以上のS37ないしS46の手順によって、すでにタイマ起動中の場合に、コネクション制御

表の更新、つなぎかえなどの処理を実行することが可能となる。

【0051】

【発明の効果】以上説明したように、本発明によれば、1つのタイマを用意し、タイマ間隔に対応づけて複数のタイマエントリを設けてタイムアウトするタイマエントリにコネクション制御表をキューイングして該当コネクション制御表をもとにタイムアウト処理をまとめて行う構成を採用しているため、多量のコネクションのタイマ処理時のそれぞれのカウントダウン処理を無くし、負荷を大幅に軽減すると共にメモリ消費量を削減することができる。

【図面の簡単な説明】

【図1】本発明のシステム構成図である。

【図2】本発明の1実施例構成図である。

【図3】本発明の動作説明フローチャート（その1）である。

【図4】本発明の動作説明フローチャート（その2）である。

【図5】本発明の動作説明フローチャート（その3）である。

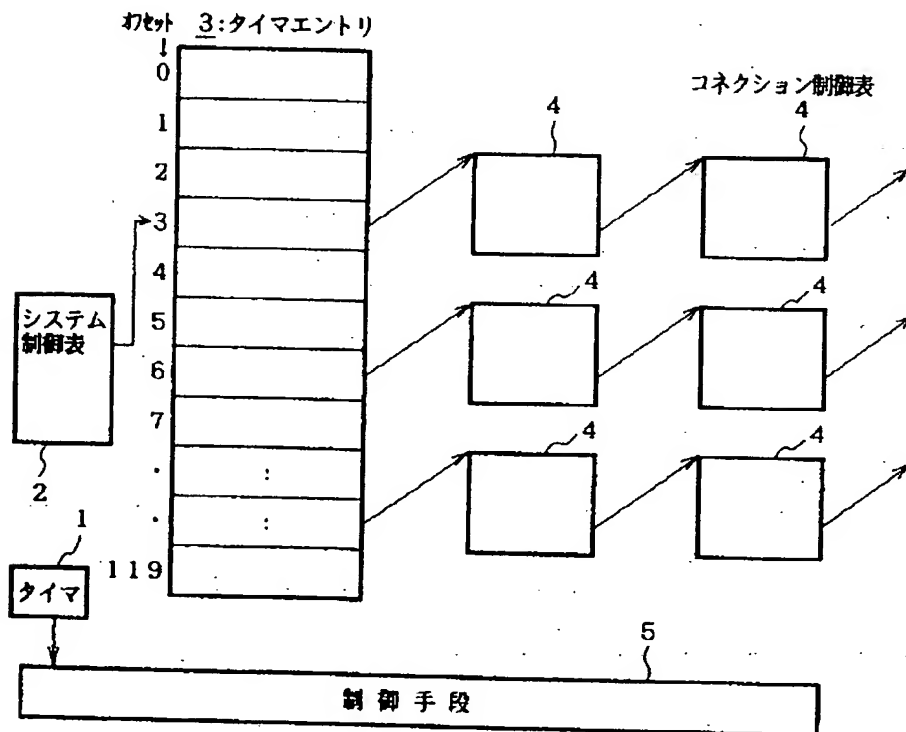
【図6】本発明の動作説明フローチャート（その4）である。

【符号の説明】

- 1：タイマ
- 2：システム制御表
- 3：タイマエントリ
- 4：コネクション制御表
- 5：制御手段

【図1】

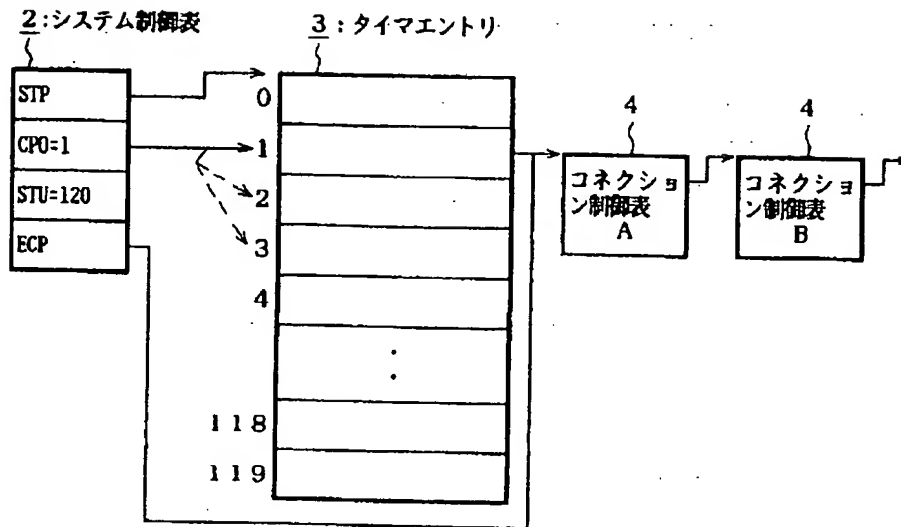
本発明のシステム構成図



【図2】

本発明の1実施例構成図

(a)



(b) システム制御表

2

システムタイマエントリ域ポインタ (STP)
現ポインタオフセット (CPO)
システムタイマエントリ数 (STU)
処理中コネクション制御表 (ECP)

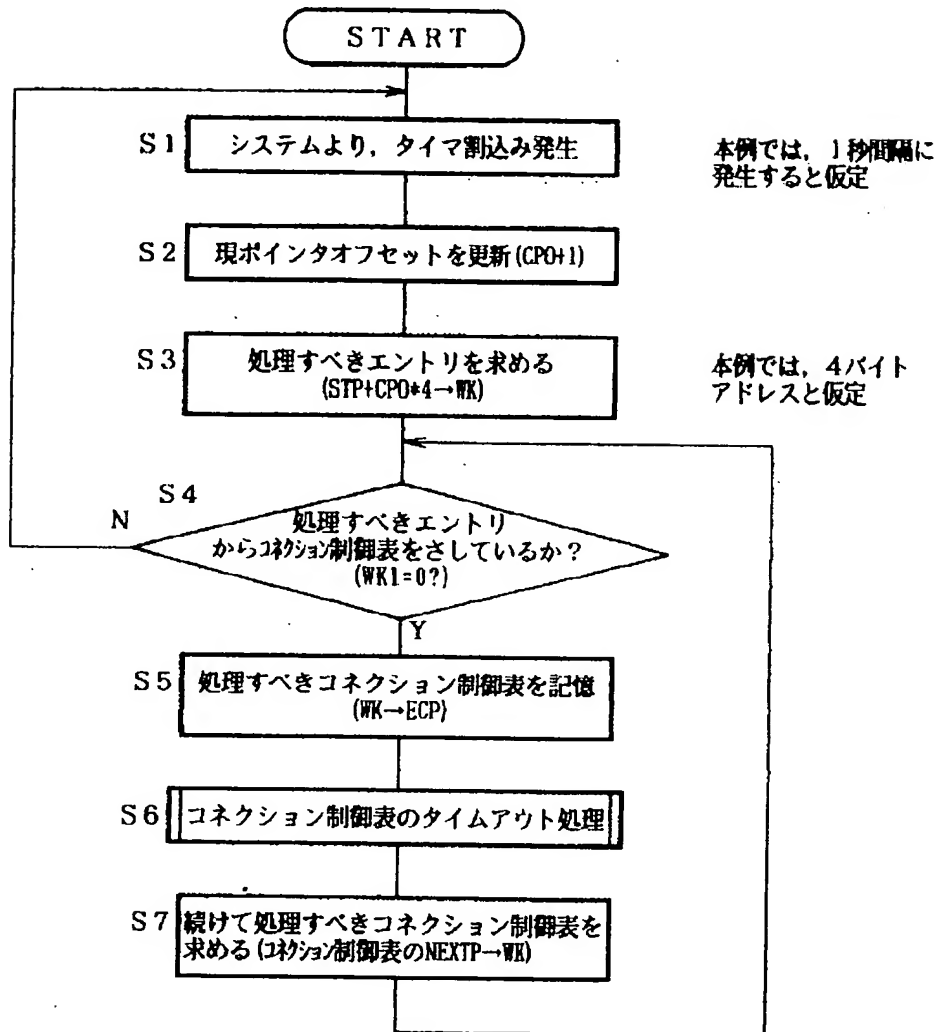
(c) コネクション制御表

4

次コネクション制御表ポインタ (NEXTP)	
タイマ起動時オフセット (TOF) *1	
タイマ1 要求値 (T1R)	タイマ1 残存値 (T1C)
タイマ2 要求値 (T2R)	タイマ2 残存値 (T2C)
タイマ3 要求値 (T3R)	タイマ3 残存値 (T3C)
タイマ4 要求値 (T4R)	タイマ4 残存値 (T4C)

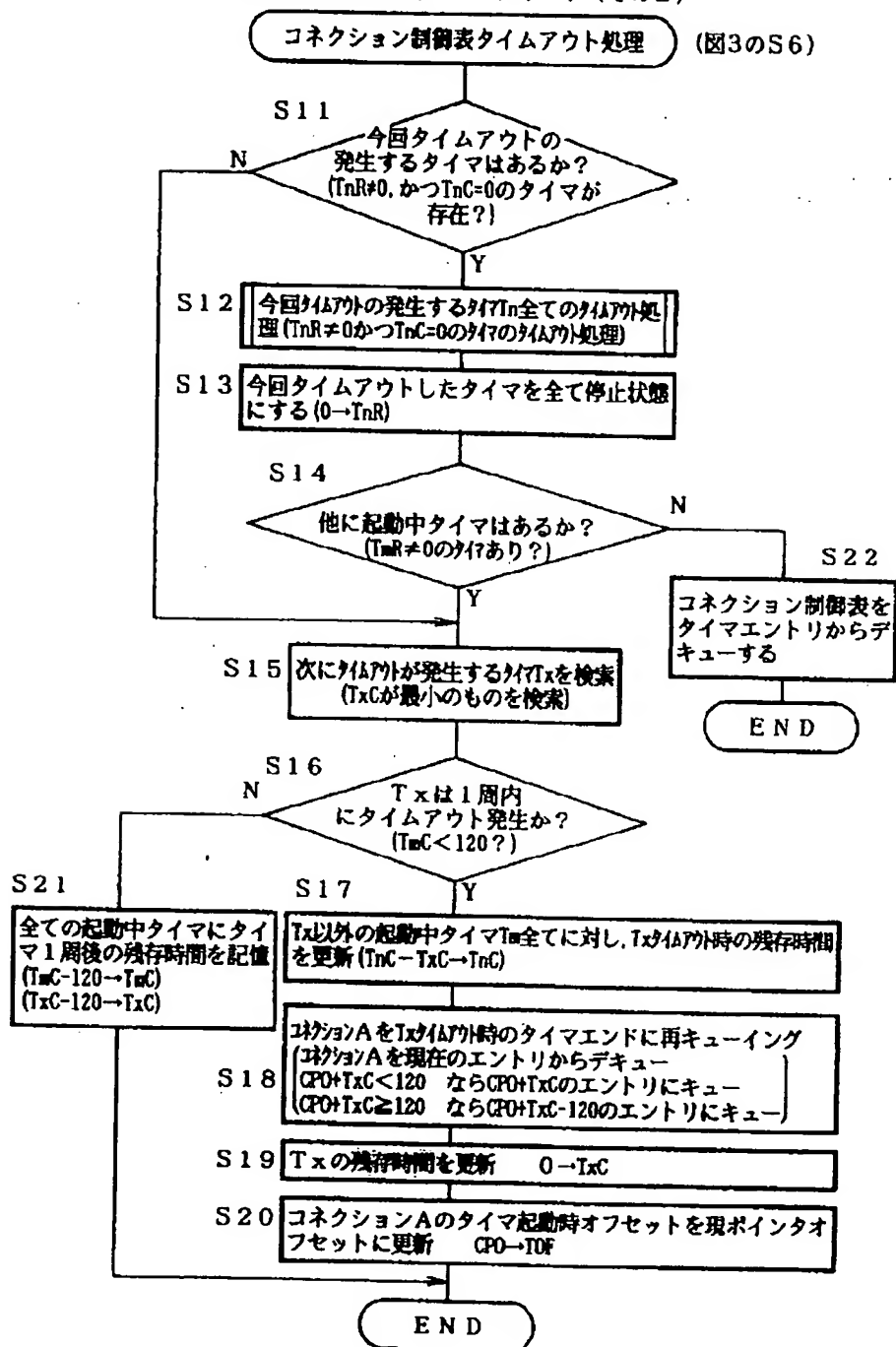
【図3】

本発明の動作説明フローチャート（その1）



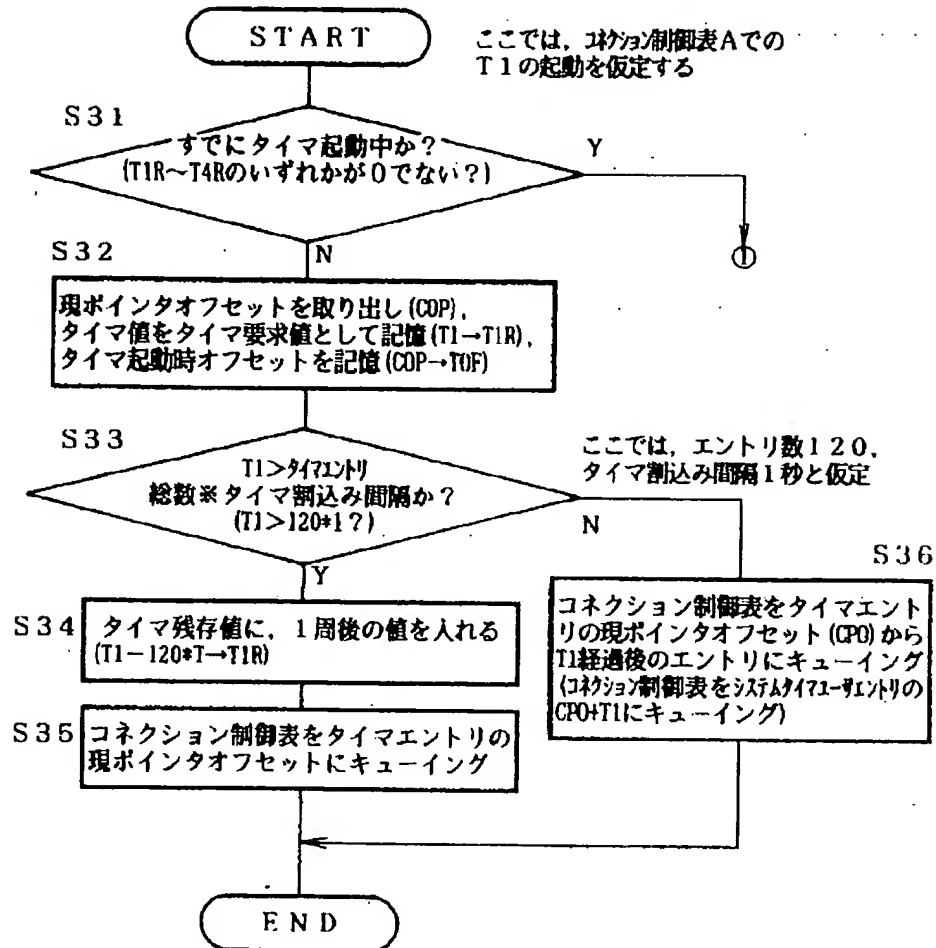
【図4】

本発明の動作説明フローチャート（その2）



【図5】

本発明の動作説明フローチャート（その3）



【図6】

本発明の動作説明フローチャート（その4）

